Efficient Search and Verification for Function Based Classification from Real Range Images

Ilan Shimshoni
Department of Management Information Systems
University of Haifa
Haifa 32000, Israel
ishimshoni@mis.haifa.ac.il

Ehud Rivlin and Octavian Soldea
Department of Computer Science
The Technion, Israel Institute of Technology
Haifa 32000, Israel
{ehudr,octavian}@cs.technion.ac.il

Abstract

In this work we propose a probabilistic model for generic object classification from raw range images. Our approach supports a validation process in which classes are verified using a functional class graph in which functional parts and their realization hypotheses are explored. The validation tree is efficiently searched. Some functional requirements are validated in a final procedure for more efficient separation of objects from non-objects. The search employs a knowledge repository mechanism that monotonically adds knowledge during the search and speeds up the classification process. Finally, we describe our implementation and present results of experiments on a database that comprises about one-hundred-and-fifty real raw range images of object instances from ten classes.

1 Introduction

The object recognition field in computer vision made its debut in the area of object identification and has moved, in recent years, towards classification. Little work has focused on the classification problem, where the imaged object is to be categorized as one of a set of classes of objects rather than a specific known object. In this context, functional based classification analyzes the function an object can fulfill. This function is in fact a classification criterion.

The possible advantages of functional approaches to generic classification were recognized early on in works such as [4]. Although several systems for object classification (see [4, 5]) were devised on the basis of these approaches, little experimental work has been done to test them. Only preliminary attempts have been made towards functional classification using raw images or stereo image pairs [11].

An impressive number of good results in the function-based classification field were demonstrated with the GRUFF, OMLET, and OPUS systems [10, 11]. GRUFF and OMLET were tested on raw images that included mock chairs built from boxes. Relatively recent specializations of functional based reasoniong were reported in the literature [1, 7, 8, 9].

We describe a general scheme for classifying objects from raw range images. In the low-level phase, we process the raw input image to obtain a segmentation into primitive shape parts: sticks, plates, and blobs, as well as the relationships between them. In the high-level phase, we employ functional part recognizers to compute mappings from primitive parts to their functionalities. We employ graph-driven verification of classes for the actual classification, and verify if the input image conforms to the generic description of each of the known classes. Each class verification involves a validation of hypotheses for the functional parts of the class.

We address the exponential complexity of the classification process by introducing a probability-based efficient search algorithm. In addition, after hypotheses for all the functional parts have been proposed, a whole_object_test using functional based reasoning is performed. We tested our algorithm on a database of one-hundred-and-fifty range images of real 3D objects. The system was able to recognize all the objects from the classes it was designed to recognize. The importance of the probability search and the verification stage are demonstrated experimentally.

2 Generic Classification by Functional Parts

Many objects have a function they are designed to fulfill, *the primary function* [3]. We decompose the primary function into several *derived functions*. These functional parts and the relationships between them are designed to realize the derived functions. The functional parts are connected by means of relationships.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

2.1 Relating Class Description to Shape

Following the RBC (recognition-by-components) concepts defined in [2], we decompose the imaged object into a set of primitive shape parts that are used for high-level shape representation. We classify primitive shape parts into three basic classes: sticks, plates, and blobs. In order to account for the variety of shapes, we also allow bending deformations of sticks and plates.

We employ *functional recognizers* to map functional parts to shape parts. Each recognizer receives sets of primitive parts in input and provides hypotheses to output. Each hypothesis is given a grade that specifies how well it conforms to its functional requirements.

2.2 Functional Verification

We represent each class as a graph, in which each node represents a functional part of the class, and the edges represent relationships between functional parts. We call this graph the functional class graph. Our classification scheme employs a bottom-up verification of each class. This verification process is performed as described in Algorithm 1.

Let FP be the set of functional parts that appear in a functional class graph FCG and PP a set of primitive parts. An object is classified if

$$\forall fp \in FP, \exists p \in PP \times FP, p = (p, fp)\,.$$

Each hypothesis h is described as a subset of $PP \times FP$.

2.3 Cost Estimation of the Search

During classification, the functional class graph is traversed, and each traversal corresponds to a path in the search tree from the root to one of its leaves. An efficient traversal of the tree is a major factor in performance speedups.

If the functional part *does* exist in the image, we expect that, on average, one-half of the realizations have to be explored until the "correct" hypothesis is found. The total amount of futile work in this case is: $\frac{1}{2}W_r n_h$, where W_r is the expected work for exploring each realization of the functional part r, and n_h is the expected number of possible realizations in the image for this functional part. The total amount of futile work in the case that the functional part was *not* found is: $W_r n_h$. Thus, the total mean futile work is therefore:

$$W_{futile} = \frac{P_+ \cdot W_r n_h}{2} + P_- \cdot W_r n_h = \left(1 - \frac{1}{2}P_+\right) W_r n_h \quad (1)$$

Algorithm 1 Functional Verification Algorithm

Input:

A set of primitive parts - PP

FCG - A functional class graph, (let FP be its set of functional parts)

map - A list of pairs $(p_1, p_2) \in PP \times FP$

A set of cues - $rep\,(ository)$

A threshold for grade - $t\,(hreshold)$

Output: A functional classification grade

FunctionalVerification(PP, map, FCG, rep, t)

```
if all the nodes in FCG have been explored then
   return (whole_object_test (PP, map, FCG, rep))
end if
grade ← 0.0
(L_fp, SL) ← UP_Algorithm (PP, FCG, map, rep, t)
(see Algorithm 2)
for all fp ∈ L_fp do
   for all hypothesis h ∈ SL (fp) do
      map_fp ← map ⋃ h
      rep_fp ← rep ∪ {new_cues}
      grade ← max (grade, FunctionalVerification (PP,
                              map_fp, FCG, rep_fp, t))
      if grade > t then
         return (grade)
      end if
   end for
end for
return (grade)
```

where P_+ is the probability that the class would be positively verified assuming the current hypothesis for a sub-object, and $P_- = 1 - P_+$. Define the *utility* as: $U\,(fp) = 1/W_{futile}$.

For n_h, an estimate is gradually refined throughout the verification process. We estimate W_r by integrating two factors: the difficulty of exploring the functional part at hand and the average depth of a class verification path after this functional part has been explored (per realization).

We assume that a class R can be verified by determining whether the imaged object indeed belongs to the class R (we denote this event by R^+ and the complementary one by R^-). We also assume that an object of class R is constructed from several functional parts, denoted as r_i. Let R_i be a sub-object consisting of the parts $r_0, \ldots, r_i$. At each phase of the verification, we have a hypothesis H^{R_i} for a sub-object. After a new functional part r_{i+1} is explored, we add hypothesis h_{i+1}. We define the event that the hypothesis H for the sub-object R_i is valid recursively: $V_{H^{R_{i+1}}} \equiv \left\{V_{H^{R_i}}, V_{h_{i+1}}, V_{h \overset{i+1}{\leftrightarrow} H}\right\}$, where $H^{R_{i+1}} \equiv H^{R_i} \bigcup h_{i+1}$. Here, $V_{H^{R_i}}$ is the event that hypothesis H realizes R_i, $V_{h_{i+1}}$ is the event that the part r_{i+1} realizes the hypothesis h_{i+1}, and $V_{h \overset{i+1}{\leftrightarrow} H}$ is the event

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

Algorithm 2 Utility and Probability Computation

Input:
PP - A set of primitive parts
FCG - A functional class graph, (let FP be its set of functional parts)
map - A list of pairs $(p, fp) \in PP \times FP$
$repository$ - A set of cues
$threshold$ - A threshold for grade
Output: L_{fp} - A list of ordered functional parts (in their decreasing futile values)
SL - A set of lists of ordered hypotheses, where $SL(fp)$ represents an ordered list of hypotheses for the functional part fp (in their decreasing P_+ values)
UP_Algorithm$(PP, FCG, map, repository, threshold)$
  $L_{fp} \leftarrow$ all the unexplored functional parts components $fp \in FP$, i.e. $\forall fp \in L_{fp}, \nexists (p, fp) \in map$
  for all $fp \in L_{fp}$ **do**
    Compute the utility $U(fp)$ by analyzing FCG
  end for
  $sort$ L_{fp} in decreasing U values of the functional parts
  for all $fp \in L_{fp}$ **do**
    $SL(fp) \leftarrow \emptyset$
    for all hypotheses $h \in SL(fp)$ of realization of $fp \in L_{fp}$ **do**
      Compute the probability P_+ of h (see Equation (2))
      if $P_+ > threshold$ **then**
        $SL(fp) \leftarrow SL(fp) \bigcup \{h\}$
      end if
    end for
    $sort(SL(fp))$ (in the P_+ decreasing values)
  end for
  return L_{fp} and SL

that the relationships between R_i and r_{i+1} that are required for validation are positively verified. The probability that the image will be positively verified is:

$$P_+ = \frac{P(V_{H'} \mid R^+)P(R^+)}{P(V_{H'} \mid R^+)P(R^+) + P(V_{H'} \mid R^-)(1 - P(R^+))}, \tag{2}$$

where $P(R^+)$ is the a priori probability for an imaged object to be an instance of the class R, and $P(H' \mid R^\pm)$ are calculated recursively using the following equations:

$$\begin{aligned} &P(V_{H'^{R_{i+1}}} \mid R) \\ &= P(V_{h^{i+1} \leftrightarrow H} \mid V_{H^{R_i}}, V_{h^{r_i}}, R) P(V_{h^{r_{i+1}}} \mid V_{H^{R_i}}, R) \\ &\quad P(V_{H^{R_i}} \mid R) \end{aligned} \tag{3}$$

where R can be R^- or R^+ and with the following starting conditions: $P(\emptyset \mid R^+) = 1$, $P(\emptyset \mid R^-) = 1$. $P(V_{h^{i+1} \leftrightarrow H} \mid V_{H^{R_i}}, V_{h^{r_i}}, R^\pm)$ and $P(V_{h^{r_{i+1}}} \mid V_{H^{R_i}}, R^\pm)$ can be estimated statistically.

2.4 Low-Level Processing – 3D Segmentation

The literature on raw range segmentation is vast. We investigated several segmentation techniques found in the literature. Although these segmentation techniques are promising, we chose instead to employ a variation on the UE algorithm [6] that distinguishes between primitive parts: it can characterize, for example, a primitive shape as a deformed or an absolute planar plate. Another factor in our choice was the algorithm's shorter run time.

Figure 1. Intensity images of some tested objects. The right-most mug and chair are non-valid objects.

2.5 High-Level Processing

We implemented and tested several generic functional parts: placeable, leg, support-to-ground, handle, container, central-axis, wing, off-ground support, and stabilizer (in air). A "in-the-air-support" is a functional part that is realized by two wings having the same surface and size. (wings are realized by plates.) In addition, the two wings share the same normal. A "stabilizer" is a surface that can provide vertical or horizontal stability. We recognize the chair functionality class to be comprised of the relationships between a seat, a seat-to-ground, and a back-support, which is optional.

3 Experimental Results

We have tested our implementation on a database that consists of one-hundred-and-fifty range images of real objects, scanned employing a Cyberware laser-based range scanner. The database includes seventy-two objects that are valid instances of classes for which we have built classifiers, forty objects that are functionally non-valid instances of these known classes (e.g., an unstable chair or an ungraspable mug), and twenty-three objects for which we did not implement classifiers (boxes and hand-tools, for example). We also tested fifteen objects for which there are classifiers; however, we did not implement the required specialized functional part recognizers because these particular objects are very rare realizations of their functional parts. Examples of intensity images of some of the tested objects are shown in Figure 1. We tested our scheme on 15, 12, 7, 7, 3, 17, 2, 2, 4, 3 valid instances of chairs, mugs, glasses, tables, plates, airplanes, rolling-pins, umbrellas, hangers, and bags, respectively. We also tested our scheme on 8, 17, 3, 5, 0, 1, 2, 2, 1, 1 non-valid instances of these classes.

Figures 2 (a) and (b) show the intensity, range images, and the segmentations for two chairs.

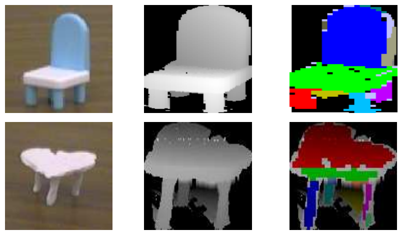

Figure 2. Chair classification.

3.1 Utility and Probability

We consider the order in which the functional parts were chosen and the order in which the realization hypotheses were explored. For each functional part, the hypotheses that were most probable were explored first. Consider, for example, the class of chairs. There, the algorithm has to decide whether to recognize the support-to-ground or the seat. In the chair in Figure 2 (a), the first part to be elaborated was the seat. In the chair in Figure 2 (b), however, the first part to be realized was the support-to-ground because the small number of leg combinations being considered led to greater utility.

We compared, for the airplane class, the time required by our classification scheme when the probabilistic mechanism described in Section 2.3 was turned on and off. When the mechanism is turned off, an arbitrary functional part is chosen by the **UP_Algorithm**. Turning the mechanism on led to a 25% speedup in classification time. We also tried an interleaved implementation. We asked if the analyzed object is one of five classes: chair, glass, mug, bed, or plate. The classifiers compete to gain processing time at the moment of proposing functional parts and hypotheses to be recognized.

3.2 The Role of Verification in Functional Reasoning

The final validation checks the interrelationships that can, naturally, only be checked in the final stages of computation, when all the functional parts have been hypothesized. For example, we tested airplanes to see if the normal of the off-ground support is perpendicular to that of the vertical stabilizer. All seventeen valid airplanes were correctly classified and their vertical stabilizer recognized. When we omitted this stage, all the airplanes were still classified, but small wings from the horizontal stabilizer were incorrectly recognized as vertical stabilizers.

Moreover, the verification stage has the added advantage of improving classification certitude. As part of our tests we modified correct objects, that were formerly recognized by our scheme, into invalid objects (for example we add an object on the seat surface) . When submitted them again to classification, the final detection of the modification took place in the whole_object_test stage.

4 Conclusions

In this paper we address the problem of object classification from raw range images. A new scheme for a functional based classification is presented. The scheme has been tested on one-hundred-and-fifty range images of real objects. The main contributions of this work are a new search method, which is implemented by means of a probability mechanism, and a verification stage, that can be performed in the last stages of computation only, when all the hypotheses about functional parts are known. We report that the recognition process speeds up when the probability mechanism is turned on.

References

[1] R. S. Amant and A. B. Wood. Tool use for autonomous agents. *AAAI*, pages 184–189, 2005.

[2] I. Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[3] L. Bogoni and R. Bajcsy. Interactive recognition and representation of functionality. *CVIU*, 62(2):194–214, 1995.

[4] M. DiManzo, E. Trucco, F. Giunchiglia, and F. Ricci. Fur: Understanding functional reasoning. *IJIS*, 4:431–457, 1989.

[5] J. Hodges. Functional and physical object characteristics and object recognition in improvisation. *CVIU*, 62(2):147–163, September 1995.

[6] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, , and R. B. Fischer. An experimental comparison of range image segmentation algorithms. *IEEE TPAMI*, 18(7):673–689, July 1996.

[7] Y. Keselman and S. Dickinson. Generic model abstraction from examples. *IEEE TPAMI*, 27(7):1141–1156, July 2005.

[8] W.-J. Li and T. Lee. Object recognition and articulated object learning by accumulative hopfield matching. *PR*, 35:1933–1948, 2002.

[9] P. Peursum, S. Venkatesh, G. A. W. West, and H. H. Bui. Using interaction signatures to find and label chairs and floors. *IEEE Pervasive Computing*, 3(4):58–65, October 2004.

[10] L. Stark and K. Bowyer. Generic object recognition using form & function. *Machine Perception and Artificial Intelligence, World Scientific*, 10, 1996.

[11] M. A. Sutton, L. Stark, and K. Hughes. Exploiting context in function-based reasoning. *Lecture Notes in CS*, 2238:357–373, 2002.

0-7695-2521-0/06 $20.00 © 2006 IEEE